

TestNG Tutorials

Contents

1. TestNG Introduction and Configuration Setup.....	2
2. TestNG Installation.....	2
3. Annotations.....	2
4. Priority.....	3
5. Assertions.....	3
6. Parameters.....	3
7. Data Providers.....	3
8. Data Provider with Excel.....	4
9. Execute Failed Test Cases.....	5
10. Cross Browser Testing.....	5
11. Parallel Execution.....	6
12. Groups.....	6
13. Reports.....	7
14. Reporter Log.....	7
15. Listeners.....	7
16. How to ignore test case.....	7
17. DependsOnMethod.....	7
18. Parellel.....	7
19. expectedExceptions.....	8

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook: <https://www.facebook.com/people/RBA-Infotech/100043552406579/>

TestNG Tutorial

1. TestNG Introduction and Configuration Setup

TestNG is a testing framework inspired from **JUnit** and **NUnit** but introducing some new functionality that makes it more powerful and easier to use

2. TestNG Installation

TestNG can be installed in **two ways**

1. Via Eclipse -> Help -> Marketplace and search for testing and install it
2. Via Eclipse -> Help -> Install New Software
 - a. Click Add button
 - b. Enter Name as "testNG" and give <https://testng.org/testng-eclipse-update-site> in Location
 - c. Select TestNG and click install and continue

Reference:

<https://www.toolsqa.com/testng/install-testng/>

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook: <https://www.facebook.com/people/RBA-Infotech/100043552406579/>

3. Annotations

Annotations	Description
@BeforeSuite	It will run before all tests inside the test suite.
@BeforeTest	It will run before any test method belonging to the classes inside the <test> tag is run.
@BeforeClass	It will be run before the first test method in the current class is invoked.
@BeforeMethod	It will be run before each test method.
@Test	Marks a class or a method as a part of the test.
@AfterMethod	It will be run after each test method.
@AfterClass	It will be run after all the test methods in the current class have been run.
@AfterTest	It will be run after all the test methods belonging to the classes inside the <test> tag have run.
@AfterSuite	It will be run after all tests in this suite have run.
@Parameter	It is used to fetch simple parameters defined in the testng.xml file.
@Optional	It will take default value if no parameter is passed or the variable is not found in the testng.xml file.
@DataProvider	It is used to supply data to a test method. It must return an Object[m][n] where m is for the number of sets of data and n is for the number of parameters to be passed.

Order of Execution:

- a. Suite - @BeforeSuite - 1 @AfterSuite - 12
- b. Test - @BeforeTest - 2, @AfterTest - 11
- a. Class - @BeforeClass - 3, @AfterClass - 10
- b. Method - @BeforeMethod - 4, 7, @AfterMethod - 6, 9
- c. @Test 1 - 5, (displayMessage1)
- d. @Test 2 - 8 (displayMessage2)
- e. -> **alphabetic order** of execution of @Test name of the methods

Please Join in below link

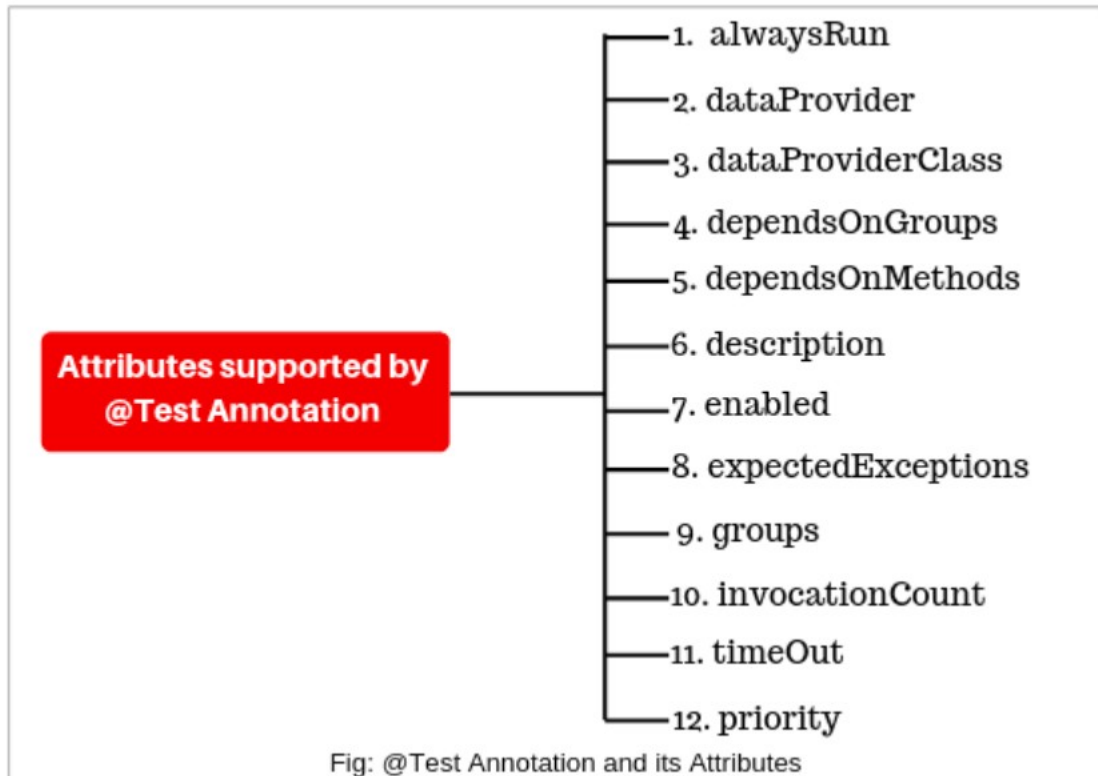
Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

4.Attributes in @Test method:



5.Priority

```
@Test(priority=1)
Public void login(){
}
```

6.Assertions

It is used to validate the test cases passed or failed

Two Types:

1. Hard Assert - Assert
2. Soft Assert - Verify

Assert.assertEquals(actualResult, expectedResult)

Assert.assertTrue(true)

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook: <https://www.facebook.com/people/RBA-Infotech/100043552406579/>

Assert.assertFalse(false)

Difference between Assert and Verify

7.How to ignore test case

f. *@Test (enabled = false).*

8.DependsOnMethod

```
import org.testng.annotations.Test;

public class DependsOnMethodsTestCase {

    @Test(dependsOnMethods = {"testCase2"})
    public void testCase1(){
        System.out.println("Test Case 1");
    }
    @Test
    public void testCase2(){
        System.out.println("Test Case 2");
    }

}
```

9.Data Providers

```
@Test(dataProvider="SearchProvider")
    public void testMethod(String author,String searchKey) throws
InterruptedException{
    {
        WebElement searchText = driver.findElement(By.name("q"));
        //search value in google searchbox
        searchText.sendKeys(searchKey);
    }
}
```

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook: <https://www.facebook.com/people/RBA-Infotech/100043552406579/>

```

        System.out.println("Welcome ->"+author+" Your search key is-
>"+searchKey);
        Thread.sleep(3000);
        String testValue = searchText.getAttribute("value");
        System.out.println(testValue + "::::"+searchKey);
        searchText.clear();
        //Verify if the value in google search box is correct
        Assert.assertTrue(testValue.equalsIgnoreCase(searchKey));
    }
}
/**
 * @return Object[][] where first column contains 'author'
 * and second column contains 'searchKey'
 */

@DataProvider(name="SearchProvider")
public Object[][] getDataFromDataprovider(){
    return new Object[][]
    {
        { "Guru99", "India" },
        { "Krishna", "UK" },
        { "Bhupesh", "USA" }
    };
}
}

```

10. Data Provider with Excel

```

FileInputStream fs = new FileInputStream("D:\\DemoFile.xlsx");
//Creating a workbook
XSSFWorkbook workbook = new XSSFWorkbook(fs);
XSSFSheet sheet = workbook.getSheetAt(0);
Row row = sheet.getRow(0);
Cell cell = row.getCell(0);
System.out.println(sheet.getRow(0).getCell(0));
Row row1 = sheet.getRow(1);
Cell cell1 = row1.getCell(1);

```

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook: <https://www.facebook.com/people/RBA-Infotech/100043552406579/>

```

System.out.println(sheet.getRow(0).getCell(1));
Row row2 = sheet.getRow(1);
Cell cell2 = row2.getCell(1);
System.out.println(sheet.getRow(1).getCell(0));
Row row3 = sheet.getRow(1);
Cell cell3 = row3.getCell(1);
System.out.println(sheet.getRow(1).getCell(1));
//String cellval = cell.getStringCellValue();
//System.out.println(cellval);
}
}

```

11. TestNG.xml file

12. Reports

1. Index.html
2. Emailable.html
3. Reporter.log();

13. Groups

```
@Test (groups = { "smokeTest", "functionalTest" })
```

TestNG.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >

<suite name="softwaretestingmaterial">
  <test name="testngTest">

    <groups>
      <run>
        <include name="smokeTest" />
      </run>

```

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook: <https://www.facebook.com/people/RBA-Infotech/100043552406579/>

```

</groups>

<classes>
<class name="softwareTestingMaterial.TestCase1" />
<class name="softwareTestingMaterial.TestCase2" />
</classes>
</test>
</suite>

```

14. Execute Failed Test Cases

Refresh the project folder

Look for “testing-failed.xml” file and execute it again after fixing the error in the failed test cases

15. Parameters

```

@Test
@Parameters ({ "val1", "val2" })
public void Sum(int v1, int v2) {
    int finalsum = v1 + v2;
    System.out.println("The final sum of the given values is " + finalsum);
}

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="TestNG Parameters Suite">
  <test name="Params">
    <parameter name="val1" value="2" />
    <parameter name="val2" value="3" />
    <classes>
      <class name="Params" />
    </classes>
  </test>
</suite>

```

16. Cross Browser Testing

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook: <https://www.facebook.com/people/RBA-Infotech/100043552406579/>


```

public class MultiBrowser {

    public WebDriver driver;

    @Parameters("browser")

    @BeforeClass

    // Passing Browser parameter from TestNG xml

    public void beforeTest(String browser) {

        // If the browser is Firefox, then do this

        if(browser.equalsIgnoreCase("firefox")) {

            //Initializing the firefox driver (Gecko)
            driver = new FirefoxDriver();

        }else if (browser.equalsIgnoreCase("chrome")) {

            //Initialize the chrome driver

            driver = new ChromeDriver();

        }

        // Enter the website address in the browser

        driver.get("https://www.demoqa.com");

    }

    // Once Before method is completed, Test method will start

    @Test public void login() throws InterruptedException {

        driver.findElement(By.xpath("//*[@id=\"app\"]/div/div/div[2]/div/div[1]/div/div[1]")).click();

    }

    @AfterClass public void afterTest() {

        driver.quit();

    }

}

```

```

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd" >
<suite name="Suite" parallel="none">

```

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook: <https://www.facebook.com/people/RBA-Infotech/100043552406579/>

```

<test name="FirefoxTest">
<parameter name="browser" value="firefox" />
<classes>
<class name="MultiBrowser" />
</classes>
</test>
<test name="ChromeTest">
<parameter name="browser" value="chrome" />
<classes>
<class name="MultiBrowser" />
</classes>
</test>
</suite>

```

17. Parallel Execution - Selenium Grid

Parallel Execution can be done with following options

1. Methods
2. Classes
3. Tests

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="softwaretestingmaterial" parallel="methods"
thread-count="2">
  <test name="testngTest">
    <classes>
      <class name="softwareTestingMaterial.ParallelTests" />
    </classes>
  </test>
</suite>

```

18. Listeners

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook: <https://www.facebook.com/people/RBA-Infotech/100043552406579/>

Listeners are TestNG annotations that literally “listen” to the events in a script and modify TestNG behaviour accordingly. Different types of Listeners are

1. *ITestListener*
2. *IReporter*
3. *ISuiteListener*
4. *IInvokedMethod*
5. *IHookable*
6. *IConfigurationListener*
7. *IConfigurableListener*
8. *IAnnotationTransformer*
9. *IExecution*
10. *IMethodInterceptor*

19. expectedExceptions

```
//package softwareTestingMaterial;  
3 import org.testng.annotations.Test;  
4 public class TestNGException {  
5  
6     @Test(expectedExceptions = ArithmeticException.class)  
7     public void testException() {  
8         System.out.println("SoftwareTestingMaterial.com");  
9         int i = 1 / 0; // the arithmetic exception is handled here because of usage
```

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook: <https://www.facebook.com/people/RBA-Infotech/100043552406579/>

```

1 expectedExceptions in      //@Test annotations)
0 }
}

```

Timeouts

```

public class TimeoutSuite
{
    @Test
    public void timeTestOne() throws InterruptedException {
        Thread.sleep(1000);
        System.out.println("Time test method one");
    }

    @Test
    public void timeTestTwo() throws InterruptedException {
        Thread.sleep(400);
        System.out.println("Time test method two");
    }
}

```

TestNG.xml

```

<suite name="Time test Suite" time-out="500" verbose="1" >
  <test name="Timeout Test" >
    <classes>
      <class name="com.howtodoinjava.test.TimeoutSuite" />
    </classes>
  </test>
</suite>

```

20. Junit vs TestNG

Basis of	JUnit	TestNG
Developed by	JUnit was developed by Kent Beck, David Saff, Erich Gamma. Erich	TestNG is a testing framework that was developed by Cédric

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook: <https://www.facebook.com/people/RBA-Infotech/100043552406579/>

Basis of	JUnit	TestNG
	Gamma, and Kris Vasudevan.	Beust.
Open-Source	JUnit is an open-source framework used to trigger and write tests.	TestNG is a Java-based framework that is an upgraded option for running tests.
Parallel test Runs	JUnit does not support to run parallel tests.	TestNG can run parallel tests.
Supports Annotation	It does not support advanced annotation.	It supports advanced annotation.
Dependency tests	The dependency tests are missing in JUnit.	Dependency tests are present in TestNG.
Grouping tests	Grouping tests together is not possible in JUnit.	Tests can be grouped together and run parallel.
Ease of Use	Running tests need a certain dependency on JUnit.	Writing tests and configuring them is easy in TestNG than JUnit.

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook: <https://www.facebook.com/people/RBA-Infotech/100043552406579/>